# Package 'cytoviewer'

September 13, 2026

**Version** 1.12.0

**Title** An interactive multi-channel image viewer for R

**Description** This R package supports interactive visualization of multi-channel images and segmentation masks generated by imaging mass cytometry and other highly multiplexed imaging techniques using shiny. The cytoviewer interface is divided into image-level (Composite and Channels) and cell-level visualization (Masks). It allows users to overlay individual images with segmentation masks, integrates well with SingleCellExperiment and SpatialExperiment objects for metadata visualization and supports image downloads.

**License** GPL-3

**Imports** shiny, shinydashboard, utils, colourpicker, shinycssloaders, svgPanZoom, viridis, archive, grDevices, RColorBrewer, svglite, EBImage, methods, cytomapper, SingleCellExperiment, S4Vectors, SummarizedExperiment

**Suggests** BiocStyle, knitr, rmarkdown, markdown, testthat

**biocViews** ImmunoOncology, Software, SingleCell, OneChannel, TwoChannel, MultiChannel, Spatial, DataImport

**VignetteBuilder** knitr

**URL** https://github.com/BodenmillerGroup/cytoviewer

**BugReports** https://github.com/BodenmillerGroup/cytoviewer/issues

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**git_url** https://git.bioconductor.org/packages/cytoviewer

**git_branch** RELEASE_3_23

**git_last_commit** 53d83b6

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-09-13

**Author** Lasse Meyer [aut, cre] (ORCID: <https://orcid.org/0000-0002-1660-1199>),
Nils Eling [aut] (ORCID: <https://orcid.org/0000-0002-4711-1176>)

**Maintainer** Lasse Meyer <lasse.meyer@dqbm.uzh.ch>

# Contents

cytoviewer . . . . . 2

**Index** **5**

---

`cytoviewer` *cytoviewer - Shiny application to interactively browse multi-channel images*

---

## Description

This shiny R application allows users to interactively visualize multi-channel images and segmentation masks generated by imaging mass cytometry and other highly multiplexed imaging techniques. The `cytoviewer` interface is divided into image-level (Composite and Channels) and cell-level visualization (Masks). It allows users to overlay individual images with segmentation masks, integrates well with `SingleCellExperiment` and `SpatialExperiment` objects for metadata visualization and supports image downloads.

## Usage

```
cytoviewer(
  image = NULL,
  mask = NULL,
  object = NULL,
  cell_id = NULL,
  img_id = NULL
)
```

## Arguments

| | |
|---|---|
| `image` | (optional) a `CytoImageList` object containing single or multi-channel `Image` objects. |
| `mask` | (optional) a `CytoImageList` containing single-channel `Image` objects. |
| `object` | (optional) a `SingleCellExperiment` or `SpatialExperiment` object. |
| `cell_id` | character specifying the `colData(object)` entry, in which the integer cell IDs are stored. These IDs should match the integer pixel values in the segmentation mask object (`mask`). |
| `img_id` | character specifying the `colData(object)` and `mcols(mask)` and/or `mcols(image)` entry, in which the image IDs are stored. |

## Value

A Shiny app object for interactive multi-channel image visualization and exploration

## The input objects

The functionality of `cytoviewer` depends on which input objects are user-provided. Below we describe the four use cases in respect to input objects and functionality.

*1. Usage of cytoviewer with images, masks and object*

The full functionality of cytoviewer can be leveraged when `image`, `mask` and `object` are provided. This allows image-level visualization (Composite and Channels), cell-level visualization, overlaying images with segmentation masks as well as metadata visualization.

*2. Usage of cytoviewer with images only*

If only `image` is specified, image-level visualization (Composite and Channels) is possible.

*3. Usage of cytoviewer with images and masks*

Image-level visualization (Composite and Channels), overlaying of images with masks and cell-level visualization is feasible when `image` and `mask` are provided.

*4. Usage of cytoviewer with masks and object*

If `mask` and `object` are specified, cell-level visualization as well as metadata visualization is possible.

## Author(s)

Lasse Meyer (<lasse.meyer@dqbm.uzh.ch>)

## See Also

`plotPixels` for the function underlying image-level visualization

`plotCells` for the function underlying cell-level visualization

`cytomapperShiny` for a shiny application that visualizes gated cells on images

## Examples

```
# Load example datasets from cytomapper
library(cytomapper, quietly = TRUE)
data("pancreasImages")
data("pancreasMasks")
data("pancreasSCE")

# 1. Use cytoviewer with images, masks and object
app <- cytoviewer(image = pancreasImages,
                  mask = pancreasMasks,
                  object = pancreasSCE,
                  img_id = "ImageNb",
                  cell_id = "CellNb")
if (interactive()) {
shiny::runApp(app, launch.browser = TRUE)
}
```

```
## Other input variations (see "The input objects" section):

# 2. Use cytoviewer with images
app_1 <- cytoviewer(image = pancreasImages)
if (interactive()) {
shiny::runApp(app_1, launch.browser = TRUE)
}


# 3. Use cytoviewer with images and masks
app_2 <- cytoviewer(image = pancreasImages,
                  mask = pancreasMasks,
                  img_id = "ImageNb")
if (interactive()) {
 shiny::runApp(app_2, launch.browser = TRUE)
}

# 4. Use cytoviewer with masks and object
app_3 <- cytoviewer(mask = pancreasMasks,
                  object = pancreasSCE,
                  img_id = "ImageNb",
                  cell_id = "CellNb")
if (interactive()) {
 shiny::runApp(app_3, launch.browser = TRUE)
}
```

# Index

cytomapperShiny, *3*
cytoviewer, 2

plotCells, *3*
plotPixels, *3*